TREX: A Temporal Reference Extractor for Arabic Texts

Ramzi A. Haraty and Samer A. Khatib
Lebanese American University
Email: rharaty@lau.edu.lb, khatibsam@hotmail.com

Abstract

Our work demonstrates the process of extracting the temporal references from the texts. This procedure is highly dependable on the stemming process. A list of all the temporal references is used. The type of the temporal word decides the procedure to treat this word and gives the importance of this temporal reference. These conditions, with the help of the stemmer, produced an excellent result of 95% as precision and of 91% as recall.

1. Intorduction

Up till recently, the work on extracting temporal references from the Arabic documents was not satisfactory. The reason for this dissatisfaction was due to bad results in the stemming process. The stemming process as we know is the backbone of extracting temporal references. The recall of the stemmer was not reaching even 65% [8]. Since Arabic is the official language of over twenty Middle Eastern and African countries, it is not acceptable to extract temporal references with a low recall stemmer. Working on these topics will have a great effect on our society and education level because it will make the path to correct data shorter. It will also save time for researchers and students alike. In addition almost all the documents we work on or use are written in one date, while describing events which occurred on different dates. So perhaps the following question is in order: how to search for a specific document describing events happening on a specific date? This is what we need to do in extracting Gregorian dates and Hijri (Arabic) dates.

The rest of the paper is organized as follows: Section 2 presents related work. In section 3, we describe the process of extracting temporal references. A list of each category of words or temporal reference terms is shown with its effect on the temporal reference string and its effect on the rank of this reference. In section 4, experimental results are shown on 25 texts. A conclusion is drawn in section 5.

2. Related Work

Numerous works were done on the temporal references extracting in different languages, but not on the Arabic language. [1] presented a new object-oriented document model, named TOODOR, which stands for Temporal Object-Oriented Document Organization and Retrieval. This temporal database model represents the properties of historical data. [2] also worked on this model and took into consideration two time dimensions: the publication date, and the event-time period of documents. TOODOR assumes that the event-time period of a document is manually assigned by specialists, which is an important limitation. This limitation was solved by [3], who extracted temporal information from document texts and translated them into temporal expressions of a formal time model. From these expressions, they were able to approximately calculate the event-time periods of documents. [4] described a semantic tagging system that extracts temporal information from news messages. Temporal expressions are defined for this system as chunks of text that express some sort of direct or inferred temporal information. Relying also on TOODOR, [5] developed a new declarative retrieval language for historical documents. The purpose of this language is to facilitate the specification of complex conditions so that historical documents can be retrieved by their contents, structure, and temporal relationships. [7] presented a tool for extracting time (temporal) references from textual documents. It employed a set of simple rules and a finite state automaton to compute time indices of documents based on their contents and reference dates. Time references extracted from texts can even help to index video content [6].

3. Extracting Temporal References

If one reads three documents written on the same day, the first document may be talking about

0-7803-8735-X/05/$20.00©2005 IEEE

an event that happened on that same day, while the second might be about things that occurred centuries ago, and the third about things that are likely to happen in the future. In this section, we will describe how we are able to retrieve the temporal reference from a document and the ability to show the importance of this temporal reference.

We added to the Stoplist table all the temporal references and the words that we believe could indicate or predict the presence of a temporal reference. Each of these temporal references has its own type that we name "TermTypeID". All the types, their meaning and some samples are shown in tables (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, and 12).

Type "30" is used for the names of the days, words as "months", "years", etc. (Table 1).

Table 1. Temporal references of type "30".

StopList				
Term	TermTypeID	DayVal	BeforeAfter	StemWord
الجمعه	30	0	0	جمع
الخميس	30	0	0	خمس
السبت	30	0	0	سبت

Type "31" has a special case. Each of these references is normally followed by either "الاول-الاولى" ("first") or "الثاني-الآخرة" ("second"). Thus, whenever any of these terms is found, one should look ahead to predict the next word (Table 2).

Table 2. Temporal references of type "31".

StopList				
Term	TermTypeID	DayVal	BeforeAfter	StemWord
جماد	31	0	0	جماد
ربيع	31	0	0	ربيع
تشرين	31	0	0	تشرين
كانون	31	0	0	كانون

Type "32" is only the word "صفر", and we added it only for the sake that this word in Arabic has two meanings:

1- The Arabic month that comes directly after "محرم" ("Moharam") is called "صفر" ("Safar").
2- The number zero.

Type "33" comprises "ظروف الزمان". They all indicate the presence of a temporal reference. For example, the word "منذ" ("from") definitely followed by a temporal reference such as: " منذ عام 1971" ("since the year 1971"). However, this type can come in the sentence as "قبل دخولك" ("before you enter"). This sentence is a temporal reference, but it is of no importance; thus, it is not extracted as temporal reference (Table 3).

Table 3. Arabic time adverb "ظروف الزمان".

StopList				
Term	TermTypeID	DayVal	BeforeAfter	StemWord
وسط	33	0	30	وسط
بدايه	33	0	30	بدى
نهايه	33	0	30	نهى

Type "34" is also a temporal reference that normally needs type "33" as a prerequisite to give a clear temporal reference. However, they are not categorized as good temporal references (e.g., " بعض الاحيان" ("some times")) (Table 4).

Table 4. Temporal references of type "34".

StopList				
Term	TermTypeID	DayVal	BeforeAfter	StemWord
ساعه	34	0	30	ساعه
احيان	34	0	0	حين
احيانا	34	0	40	حين
الان	34	0	30	الان

Type "35" is also another temporal reference, which is clearer and more powerful than type "34", even if it does not give an exact time reference, as when we say "حوالي الساعة 10", which means "about ten o'clock" (Table 5).

Table 5. Temporal references of type "35".

StopList				
Term	TermTypeID	DayVal	BeforeAfter	StemWord
اثناء	35	0	0	ثنى
حوالي	35	0	40	حوالي
عند	35	0	30	عند

Type "36" is the season name, and of words like: year, month, term etc. (Table 6).

Table 6. Temporal references of type "36".

StopList				
Term	TermTypeID	DayVal	BeforeAfter	StemWord
سنوي	36	0	0	سنة
شتاء	36	0	0	شتاء
خريف	36	0	0	خريف

Type "37" includes all the names of the months: Arabic months and Arabic Gregorian months (Table 7).

Table 7. Arabic months and Arabic Gregorian months.

StopList				
Term	TermTypeID	DayVal	BeforeAfter	StemWord
يناير	37	0	0	يناير
اذار	37	0	0	اذار
رجب	37	0	0	رجب

Type "40" comprises numbers: one, two, three, thousand, etc. (Table 8).

Table 8. Numbers.

StopList				
Term	TermTypeID	DayVal	BeforeAfter	StemWord
اربع	40	0	0	ربع
الف	40	0	0	الف
عاشر	40	0	0	عشر

Type "46" contains the words that clarify and describe the temporal reference; for example, " في اليوم التالي" ("the next day"). These temporal references cannot stand alone; rather, there should be a temporal reference of higher priority (types "2" or "3") for these to be attached to it. In the previous example, the word "اليوم" ("day") is the word of higher priority which is attached to the word "التالي" (Table 9).

Table 9. Temporal references of type 46.

StopList				
Term	TermTypeID	DayVal	BeforeAfter	StemWord
ظلام	46	0	0	ظلم
نصف	46	0	0	نصف
تالي	46	0	0	تلى

Type "47" comprises the same characteristics of words of type "46". But here they can stand alone as a temporal reference (Table 10).

Table 10. Temporal references of type 47.

StopList				
Term	TermTypeID	DayVal	BeforeAfter	StemWord
قدم	47	0	0	قدم
مضى	47	0	0	مضى
حالي	47	0	0	حال

There are also some words that imply the presence of a temporal reference such as " احرف الجر" as in (Tables 11 and 12).

Table 11. Words that indicate the presence of a temporal reference.

StopList				
Term	TermTypeID	DayVal	BeforeAfter	StemWord
الى	3		20	
ففي	3		30	
من	3		10	

Table 12. Other words that indicate the presence of a temporal reference.

StopList				
Term	TermTypeID	DayVal	BeforeAfter	StemWord
كذا	10		0	
بضعه	10		0	
بضع	10		0	

Finally, type "45" contains one word "ذو". When used alone, as in the sentence "ذو علم" ("has knowledge"), this word will not be a temporal reference. However, if followed by "الحجة" or "القعدة", it is treated as a temporal reference which is an Arabic month "ذو الحجة" ("Zou Al Hijja") or " ذو القعدة" ("Zou Al Kieda").

As it is clear from the preceding samples, we have the field "StemWord", which is the correct stemming of the word. Since we are searching for these references, we can directly take this "StemWord" and add it to the stemmed words instead of sending them to the stemmer to be stemmed.

After passing through a special routine that indicates whether or not this word is a StopList word, the variable "IsTerm" returns the result. If it is a StopList variable, we have two options:

1. Temporal reference or a word that indicates the presence of a temporal reference, or

2. Meaningless StopList.

Our concern is the first type where the TermTypeID is one of these {3, 10, 30, 31, 32, 33, 34, 35, 36, 37, and 40}. The following routine describes how we handled some of these types:

If the TermTypeID = {3, 10} the following condition is activated:

```
'The following condition was added only
to print a temporal reference before
starting to gather the new one, which will
start by entering the previous condition. It
is also used to delete any vague temporal
reference as "في", which could be due to
previous pass through this condition and
not finding any temporal reference other
than this word "حرف الجر"

If TimeRef <> 0 And WordedYear <> 0
And TemporalRef <> "" Then
        TempToBePosted = TemporalRef
        PostTemporalRef = True
        TemporalRef = cleanword 'we
    put this word in the temporal reference
    field.
        TimeRef = 1 'this is an indicator
        which means that we have a
        temporal reference data in
                the temporal
        reference field.
    Endif

    If PrecededTermType = 30 Or 32 Or 34 Or
37 Then
        ExtractedTerm =
    Terms.AddTermOnly(Document.Item(i).
    Word, StopListStemWord)
        'AddTermOnly is a procedure
        that adds the StemWord, which is
        retrieved directly from
         the StopList table without going
        into the stemming process.

        'The helow condition is to show
        the importance of the temporal
        reference. It shows that
         the type 37 is a very important
        temporal reference while 30 is a
        moderate one.

    If PrecededTermType = 30 Then
tempBMG = 2

    If PrecededTermType = 37 Then
tempBMG = 3

        'the condition below is applied if there is a
        year of format 9999 after the above types
        which is a kind of looking ahead; and that
        was why we used the "i+1" instead of "i"
        in the Document.Item(i + 1).Word .

        TemporalRef = TemporalRef & " " &
cleanword
    If i < cnt Then 'this condition is to ensure
that we did not reach the end of the file.

    If IsNumeric(CharAt(Document.Item(i +
1).Word, 1)) Then
            TimeRef = 1
```

The following sample of the temporal references that we extracted:

- في 10 تموز1971 ; من العام 1939
- في يناير ; في شهر ذو الحجة سنة 1423
- سنة الف وتسعماية وخمسة وعشرون ; الساعة الحادية عشرة

These and many others are clearly marked and added as a temporal reference (with its importance). There are also the formatted dates and times as we have called them. These are the dates of different formats, with the following sample:

- dd/mm/yy
- dd/mm/yyyy
- dd-mm-yyyy
- dd/mmmm/yyyy
- mm/dd/yyyy
- hh:mm:ss
- etc…

These types of dates are very important, and they are rated as an important temporal reference. They are handled in the following procedure:

```
    If Not IsNumeric(firstlet) Then
            TemporalFormat =
    TemporalTerm.WhatTemporalFormat(Tri
    m(Document.Item(i).Word),
    MyFormatType)
            If (MyFormatType = 1 Or
MyFormatType = 2) Then tempBMG = 3
                    MyFormatType = 0
```

```
                If TimeRef <> 0 Or WordedYear
<> 0 Then
                        TemporalRef =
TemporalRef & " " & cleanword
                        End if
                End If
        End if
```

The "WhatTemporalFormat" function returns the format type of the temporal reference through a special function called "GetFormat". If the format belongs to the array "lstDateFormat" that contains all type of date formats or to the array "lstTimeFormat" that contains the entire time formats, then it is considered to be a good temporal reference.

Needless to mention that if the format "99/99/9999" was found in the document as "99 / 99 / 9999" or "99 – 99 – 9999", it is also taken into consideration and combined as a temporal reference.

The importance of the reference is affected by the data (temporal references) the reference contains. The variable "tempBMG" is responsible for determining the importance of the temporal reference. It has three values:

- "1" for poor references as "في التسعينيات".
- "2" for medium references as " من خمس سنوات".
- "3" for strong references as "1971 عام".

4. Experimental Results

In this paper, we describe the process of testing the time extractor. We used 25 different texts unaltered, that is, without touching the structure of the text. The texts were of different lengths. The following terminology was used to clarify the experiment:

1- L: length of the text.
2- MR: number of temporal references retrieved manually.
3- CR: number of temporal references retrieved correctly by the computer.
4- RT: number of temporal references retrieved correctly but given a wrong type. For example, "1998 في" was given type "1".
5- NR: number of temporal references that was not retrieved.
6- RNC: number of temporal references that was retrieved but was not complete. For example, the temporal reference "في العصر الحجري" was retrieved "في العصر".
7- MAR: the number of temporal references that are ambiguous but retrieved manually. For example, "عام" means "year" and "general". So when retrieved and means "general", it is considered as ambiguous temporal reference.
8- CAR: number of temporal references that are ambiguous but retrieved by the computer.

To evaluate the results, we include the two usual metrics as in the stemming testing: precision and recall. Precision measures the percentage of relevant results - how well the retrieval algorithm avoids returning results that are not relevant; i.e., precision = CR/ (CR + NR). Recall measures the completeness of retrieval of relevant temporal references. That is Recall = MR / CR.

The results show a precision of 95% and a recall of 91%. The detailed result is shown in table 13.

Table 13. Temporal references extracting results.

Text#	L	MR	CR	CAR	MAR	NR	RT	RNC	Recall	Precision
1	467	3	3	2	2	0	1	0	1.00	1.00
2	1044	19	14	5	5	3	2	0	0.74	0.82
3	789	20	17	4	4	3	0	0	0.85	0.85
4	532	7	6	4	4	1	0	0	0.86	0.86
5	532	11	11	0	0	0	0	0	1.00	1.00
6	546	6	5	1	1	0	2	0	0.83	1.00
7	587	12	10	2	2	2	1	0	0.83	0.83
8	887	27	22	1	1	2	3	0	0.81	0.92
9	1039	21	21	2	2	0	2	0	1.00	1.00
10	846	12	11	1	1	1	0	0	0.92	0.92
11	1023	51	45	5	5	2	4	0	0.88	0.96
12	714	13	11	2	2	2	0	0	0.85	0.85
13	767	25	22	1	1	2	1	0	0.88	0.92
14	513	4	4	0	0	0	0	0	1.00	1.00
15	413	6	6	2	2	0	0	0	1.00	1.00
16	424	4	4	5	5	0	0	0	1.00	1.00
17	463	6	6	2	2	0	0	0	1.00	1.00
18	1231	22	21	0	0	0	1	0	0.95	1.00
19	577	6	6	4	4	0	0	0	1.00	1.00
20	559	6	6	2	2	0	0	0	1.00	1.00
21	496	10	8	0	0	0	0	2	0.80	1.00
22	451	13	12	6	6	0	2	1	0.92	1.00
23	599	12	9	3	3	1	2	0	0.75	0.90
24	633	7	7	0	0	0	0	0	1.00	1.00
25	720	25	21	2	2	0	3	1	0.84	1.00
Average									0.91	0.95

We did not find the results of previous work on extracting temporal references for Arabic language to compare with the result of our work. However, we can show some results of extracting temporal references in English language keeping in mind the simplicity of English language as compared to the complexity of Arabic. [4] showed a precision of 87.30% and a recall of 90.66%. [3] also showed a better performance upon analyzing four newspapers containing 1,634 time expressions. The overall precision (valid extracted dates / total extracted dates) of the evaluated set was 96.2% while the overall recall (valid extracted dates / valid dates in the set) was 95.2%.

5. Conclusion

We implemented a procedure that extracts the temporal references from Arabic texts. This procedure is highly dependable on the stemming process. A list of all the temporal references is used. This list is a part of the stop list words, and every word that was stemmed was directly checked to verify if it belongs to that list. The type of the temporal word decides the procedure to treat this word and gives the importance of this temporal reference. These conditions, with the help of the stemmer, produced an excellent result of 95% as precision and of 91% as recall.

Further research includes adding more temporal word references to the stop list words with the correct type.

Further work should also be done on assigning the text a correct time reference after extracting the temporal references from it.

References

[1] Aramburu, M. J., and Berlanga, R. *Temporal Object-Oriented Document Organization and Retrieval.* Integrated Design and Process

Technology, Volume 2: Issues and Applications of Database Technology. Ed. Society for Design and Process Science, Berlin, July 1998, pp 368-375.

[2] Aramburu, M. J., and Berlanga, R. "Retrieval of Information from Temporal Document Databases." First Workshop on Object Oriented Databases, In Conjunction with the European Conference in Object Oriented Programming, Lisboa, June 1999, pp. 85-95.

[3] Llido, D, Berlanga, R., and Aramburu, M. "Extracting Temporal References to Assign Document-Event Time Periods." In: *DEXA 2001 Conference Proceedings*, Mayr H et al. (Eds), Springer Verlag, LNCS 2113, Berlin Heidelberg, 2001, pp. 62-71.

[4] Schilder, F, and Habel, C. From Temporal Expressions to Temporal Information: Semantic Tagging of News Messages. In: *Proceedings of ACL'01 Workshop on Temporal and Spatial Information Processing*, Toulouse, France, 2001, pp 65-72.

[5] Aramburu, M. J., and Berlanga, R. "A Retrieval Language for Historical Documents." Ninth International Conference on Database and Expert System Applications, LNCS 1460, Springer Verlag, 1998, pp. 216-225.

[6] Salway, and Tomadaki. "Temporal Information in Collateral Texts for Indexing Moving Images." In: *Proceedings of LREC 2002 Workshop on Annotation Standards for Temporal Information in Natural Language*. Setzer, A. and Gaizauskas, R. (editors), Spain, 2002, pp. 36-43.

[7] Abramowicz, W., Kaczmarek, T., Kalczyński, P., and Węcel, K. "Time-indexer: A Tool for Extracting Temporal References from Text Documents. In: *The 14th Information Resources Management Association International Conference*, Philadelphia, Pennsylvania. 2003.

[8] Khoja, S. 2001. *APT: Arabic Part-of-Speech Tagger*. URL: http://archimedes.fas.harvard.edu/mdh/arabic/NAACL.pdf [10 March 2003].

Biographies

Ramzi A. Haraty is an associate professor of Computer Science at the Lebanese American University - Beirut, Lebanon. He received his B.S. and M.S. degrees in Computer Science from Minnesota State University - Mankato, Minnesota, and his Ph.D. in Computer Science from North Dakota State University - Fargo, North Dakota. His research interests include database management systems, artificial intelligence, and multilevel secure systems engineering. He has well over 50 book, journal and conference paper publications. He is a member of Association of Computing Machinery, Arab Computer Society, and International Society for Computers and Their Applications.

Samer Khatib is a graduate student at the Lebanese American University.